

# 알튜비튜

## 최단 경로

간선에 가중치가 있는 그래프가 주어질 때, 정점 사이의 최단 경로를 구하는 알고리즘입니다.  
대표적으로 [다익스트라](#), [플로이드-워셜](#), [벨만-포드 알고리즘](#)이 있습니다.

코딩테스트에 자주 나오진 않지만 한 번 나오면 난이도 있는 문제로 나오곤 해요.

## 2021 KAKAO BLIND RECRUITMENT : 합승 택시 요금 - Level 3

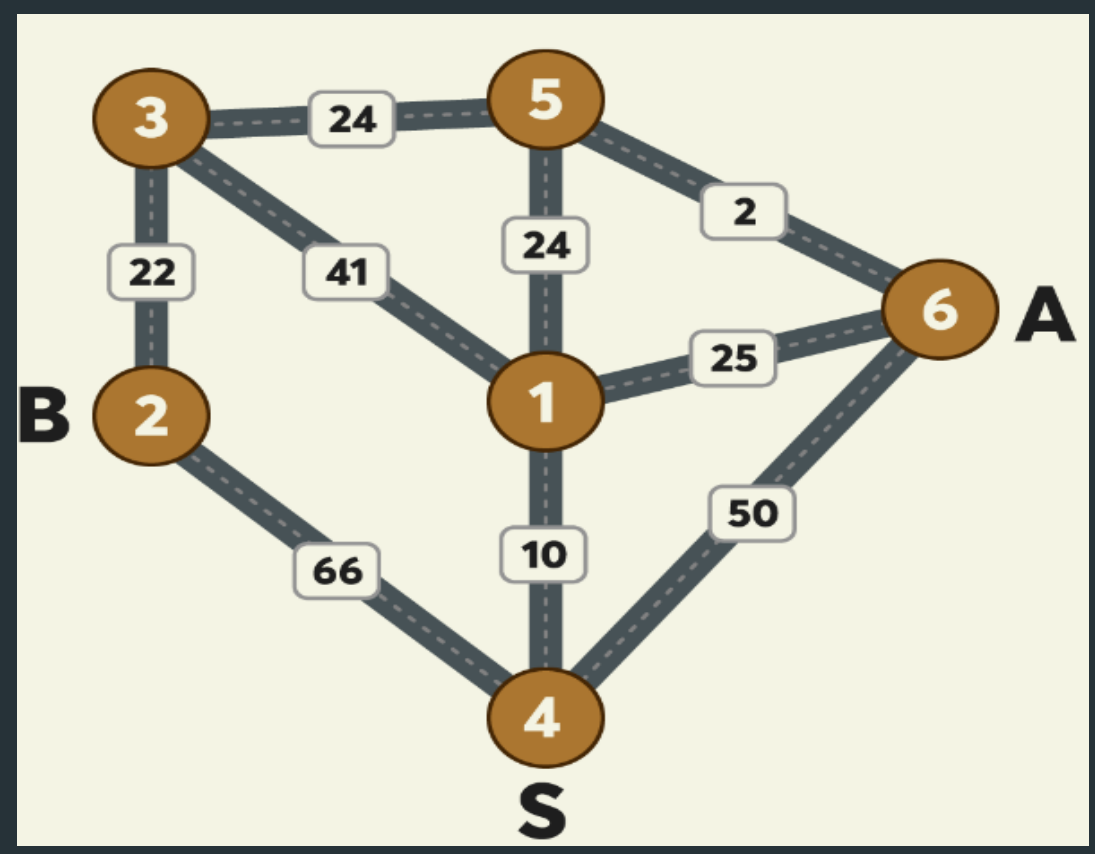
### 문제

- 지점의 개수  $n$ , 출발지점  $s$ , A의 도착지점  $a$ , B의 도착지점  $b$ , 지점 사이의 예상 택시 요금 fares일 때,
- A, B 두 사람이  $s$ 에서 출발해서 각각의 도착지점까지 택시를 타고 갈 때, 최저 예상 택시 요금을 return
- 단, 아예 합승을 하지 않고 각자 이동하는 경우가 택시요금이 더 낮다면 합승을 하지 않아도 된다.

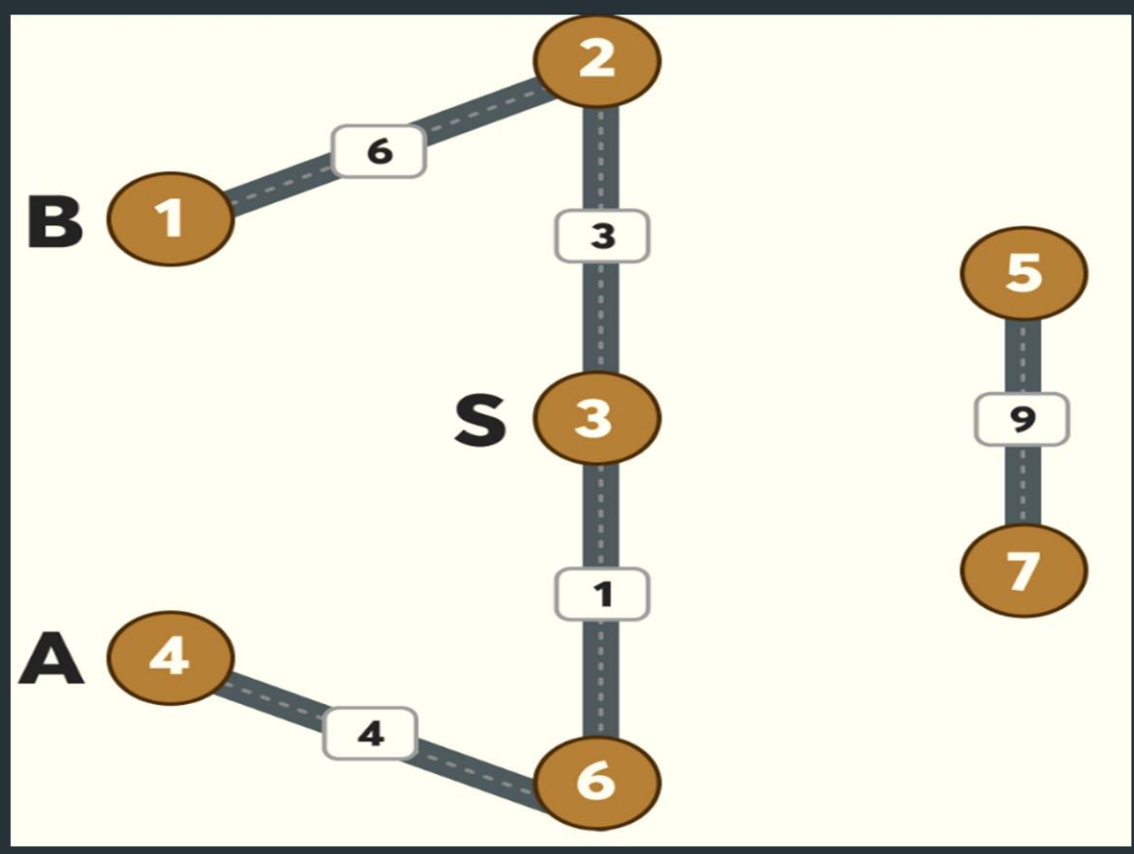
### 제한 사항

- $3 \leq n \leq 200$  이하인 자연수
- $s, a, b$ 는 1이상  $n$ 이하인 자연수이며, 각기 서로 다른 값
- fares는 2차원 정수 배열

## 예제



합승 구간 : 4 -> 1      최소 비용 : 10  
A 단독 구간 : 1 -> 6      최소 비용 : 25  
B 단독 구간 : 1 -> 3 -> 2      최소 비용 : 41 + 22 = 63  
  
총 비용 : 10 + 25 + 63 = 98



합승 구간 : X  
A 단독 구간 : 3 -> 6 -> 4      최소 비용 : 1 + 4 = 5  
B 단독 구간 : 3 -> 2 -> 1      최소 비용 : 3 + 6 = 9  
  
총 비용 : 5 + 9 = 14

## 2021 KAKAO BLIND RECRUITMENT : 합승 택시 요금 - Level 3

- A, B 두 사람이 s에서 출발해서 i ( $1 \leq i \leq n$ )지점까지 택시를 탈 때,
- $f(x \rightarrow y) :=$  지점 x에서 지점 y로 가는 최소 비용
- 최저 예상 택시 요금 =  $f(s \rightarrow i) + f(i \rightarrow a) + f(i \rightarrow b)$
- $1 \leq n \leq 200$ 이므로 플로이드-워셜 알고리즘을 사용하여 모든 정점 간의 최소 비용을 구해주면 된다

## /<> 1865번 : 웜홀 - Gold 3

### 문제

- 월드나라에는  $N$ 개의 지점이 있고  $N$ 개의 지점 사이에는  $M$ 개의 도로와  $W$ 개의 웜홀이 있다.
- 웜홀은 시작 위치에서 도착 위치로 가는 하나의 경로인데, 특이하게도 도착을 하게 되면 시작을 하였을 때보다 시간이 뒤로 가게 된다.
- 한 지점에서 출발을 하여서 여행을 한 뒤 다시 출발을 하였던 위치로 돌아왔을 때, 출발을 하였을 때보다 시간이 되돌아가 있는 경우가 있는지 없는지 확인

### 제한 사항

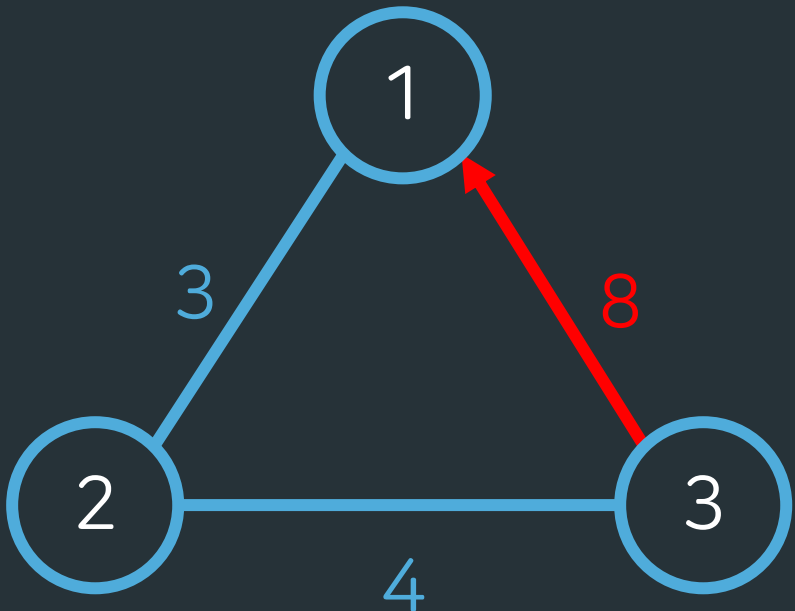
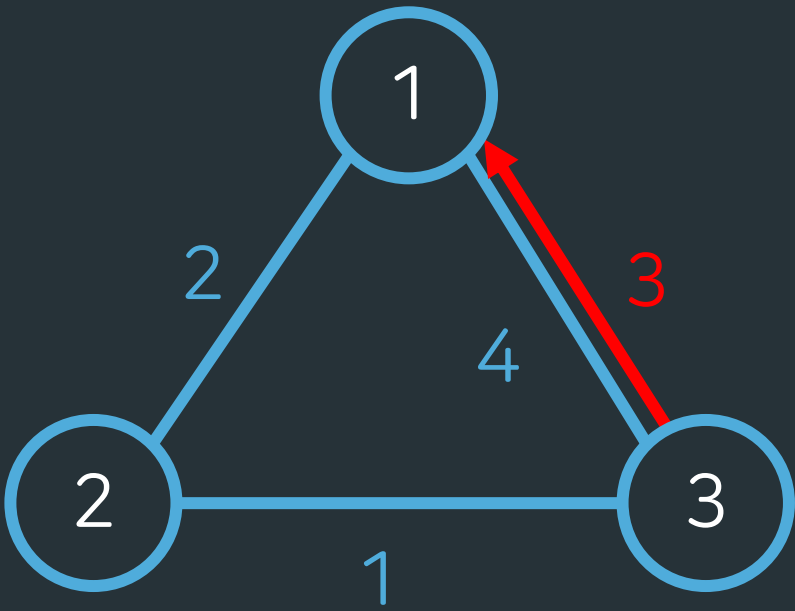
- 테스트케이스의 개수  $TC(1 \leq TC \leq 5)$ , 지점의 수  $N(1 \leq N \leq 500)$ , 도로의 개수  $M(1 \leq M \leq 2500)$ , 웜홀의 개수  $W(1 \leq W \leq 200)$
- $S$ 와  $E$ 는 연결된 지점의 번호,  $T$ 는 이 도로를 통해 이동하는데 걸리는 시간을 의미한다.
- 웜홀의 정보가  $S, E, T$  세 정수로 주어지는데  $S$ 는 시작 지점,  $E$ 는 도착 지점,  $T$ 는 줄어드는 시간을 의미한다.  $T$ 는 10,000보다 작거나 같은 자연수 또는 0이다.

## 예제 입력1

```
2
3 3 1
1 2 2
1 3 4
2 3 1
3 1 3
3 2 1
1 2 3
2 3 4
3 1 8
```

## 예제 출력1

```
NO
YES
```



시간이 줄어들면서 출발 위치로 돌아오는 것이 가능한가?

NO

YES

$1 \rightarrow 2 \rightarrow 3 = 3 + 4 - 8 = -1 < 0$

웜홀: 시간이 뒤로 간다 => 음의 간선이 존재한다

음의 간선이 있는지 확인하는 알고리즘은

벨만-포드 알고리즘

## /<> 15685번 : 드래곤 커브 - Gold 4

### 문제

- 드래곤 커브는 시작점, 시작 방향, 세대 세 가지 속성을 가지고 이차원 평면에 정의됨
- $K(K > 1)$  세대 드래곤 커브는  $K-1$  세대 드래곤 커브를 끝 점 기준 90도 시계 방향으로 회전시켜 그것을 끝점에 붙여 얻음
- 끝 점은 시작 점에서 선분을 타고 이동했을 때 가장 먼 거리에 있는 점
- 격자 위에 크기가  $1 \times 1$ 인 정사각형의 네 꼭짓점이 모두 드래곤 커브의 일부인 정사각형의 개수는?



## /<> 15685번 : 드래곤 커브 - Gold 4

### 제한 사항

- 격자의 크기는  $100 \times 100$
- 드래곤 커브의 개수의 범위는  $1 \leq N \leq 20$
- 드래곤 커브의 시작점의 좌표  $x, y$ 의 범위는  $0 \leq x, y \leq 100$
- 드래곤 커브의 시작 방향( $d$ )과 세대( $g$ )의 범위는  $0 \leq d \leq 3, 0 \leq g \leq 10$
- 드래곤 커브는 격자 밖으로 벗어나지 않으며 서로 겹칠 수 있음
- 드래곤 커브의 방향
  - 0:  $x$  좌표가 증가하는 방향 ( $\rightarrow$ )
  - 1:  $y$  좌표가 감소하는 방향 ( $\uparrow$ )
  - 2:  $x$  좌표가 감소하는 방향 ( $\leftarrow$ )
  - 3:  $y$  좌표가 증가하는 방향 ( $\downarrow$ )

## 예제 입력1

```
3
3 3 0 1
4 2 1 3
4 2 2 1
```

## 예제 출력1

4

## 예제 입력2

```
4
3 3 0 1
4 2 1 3
4 2 2 1
2 7 3 4
```

## 예제 출력2

11

## 예제 입력3

```
10
5 5 0 0
5 6 0 0
5 7 0 0
5 8 0 0
5 9 0 0
6 5 0 0
6 6 0 0
6 7 0 0
6 8 0 0
6 9 0 0
```

## 예제 출력3

8

## 예제 입력4

```
4
50 50 0 10
50 50 1 10
50 50 2 10
50 50 3 10
```

## 예제 출력4

1992

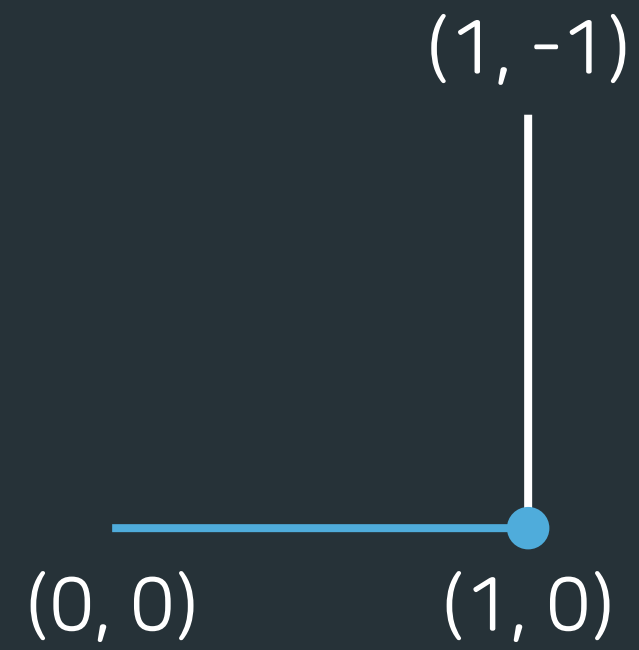
# 드래곤 커브를 그려보자



$(0, 0)$   $(1, 0)$

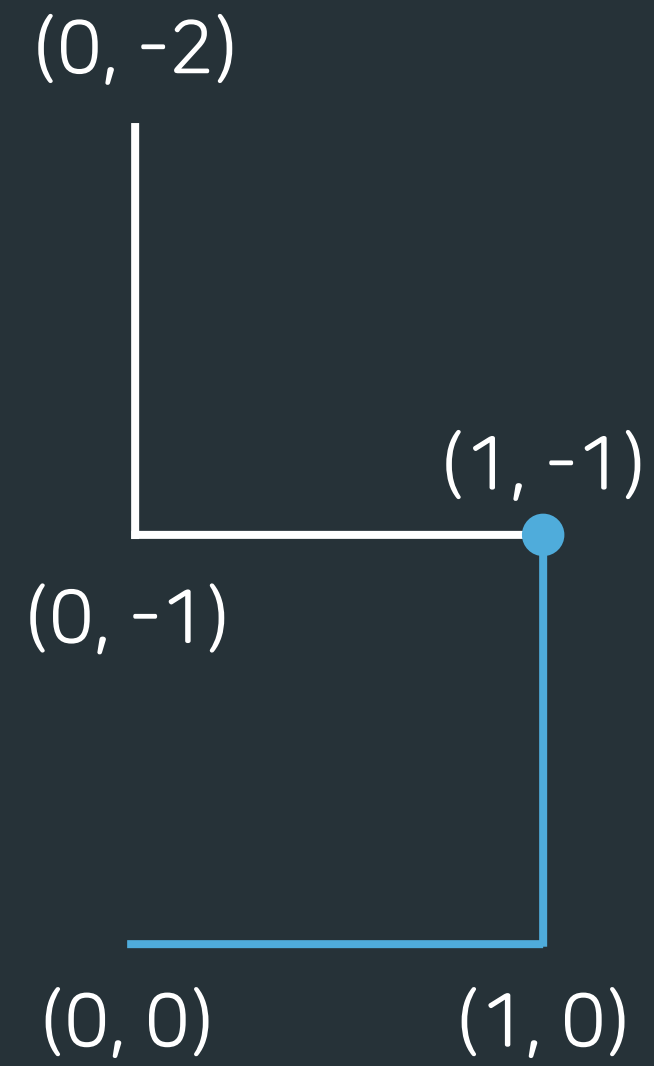
0세대

# 드래곤 커브를 그려보자



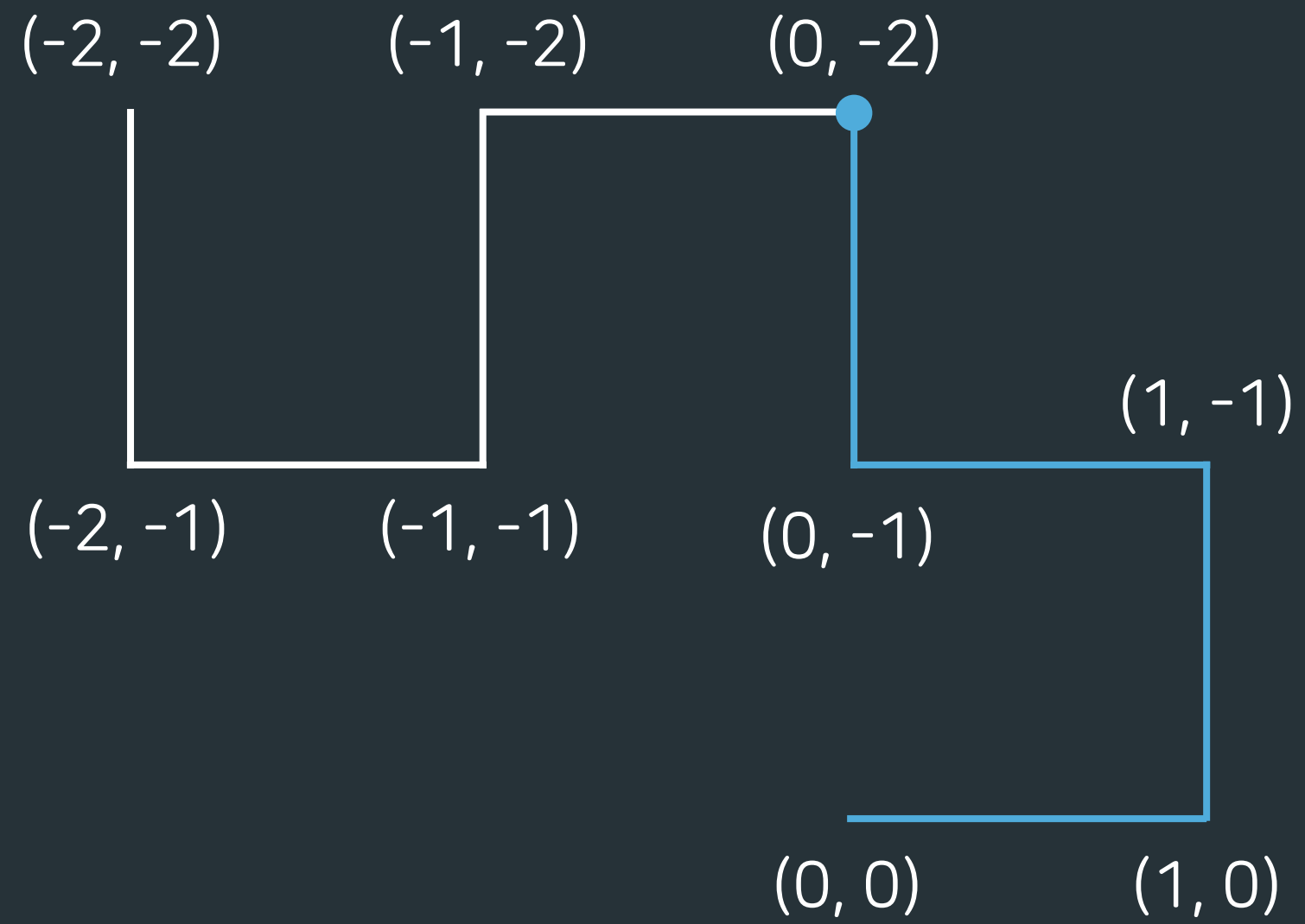
1세대

# 드래곤 커브를 그려보자



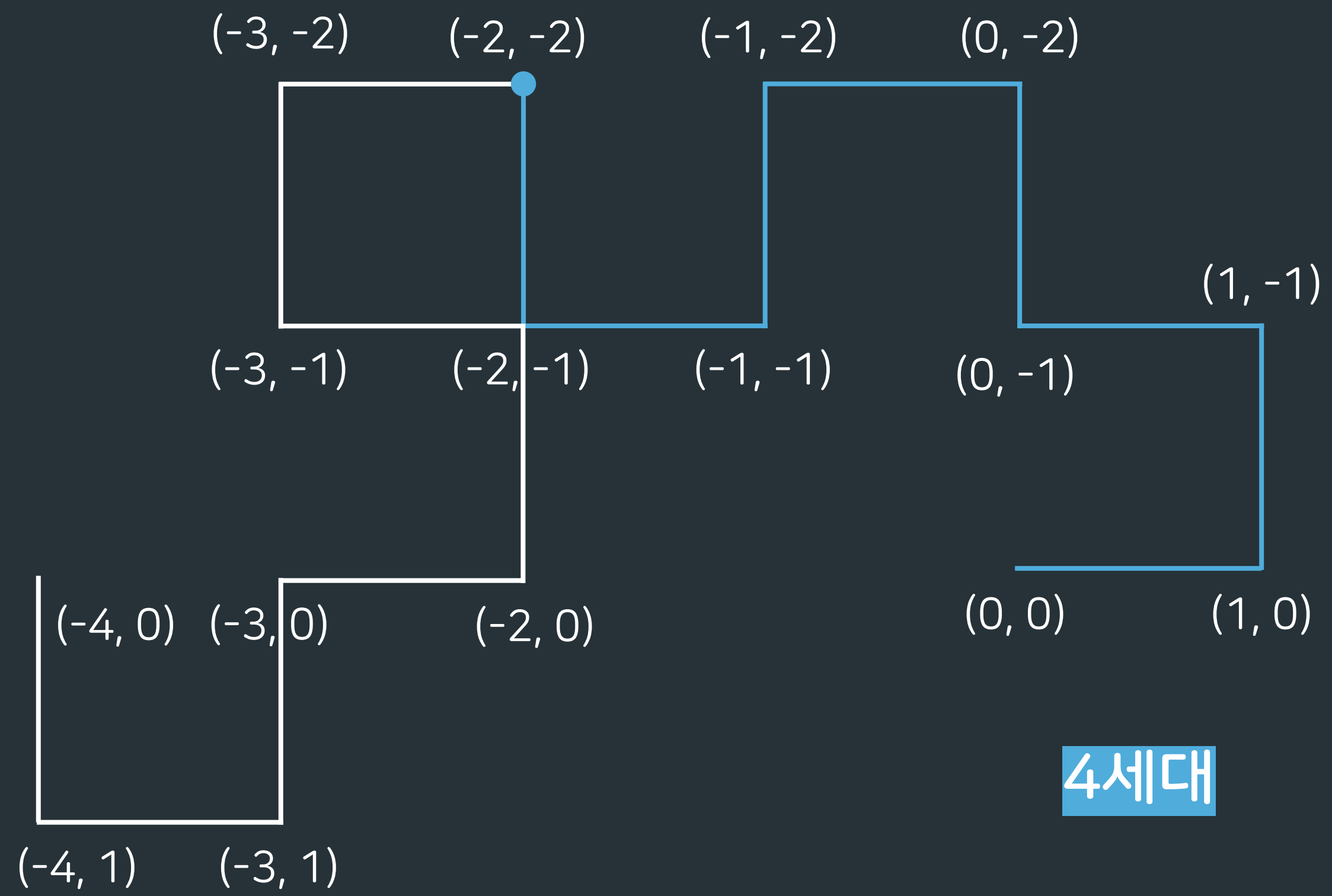
2세대

# 드래곤 커브를 그려보자

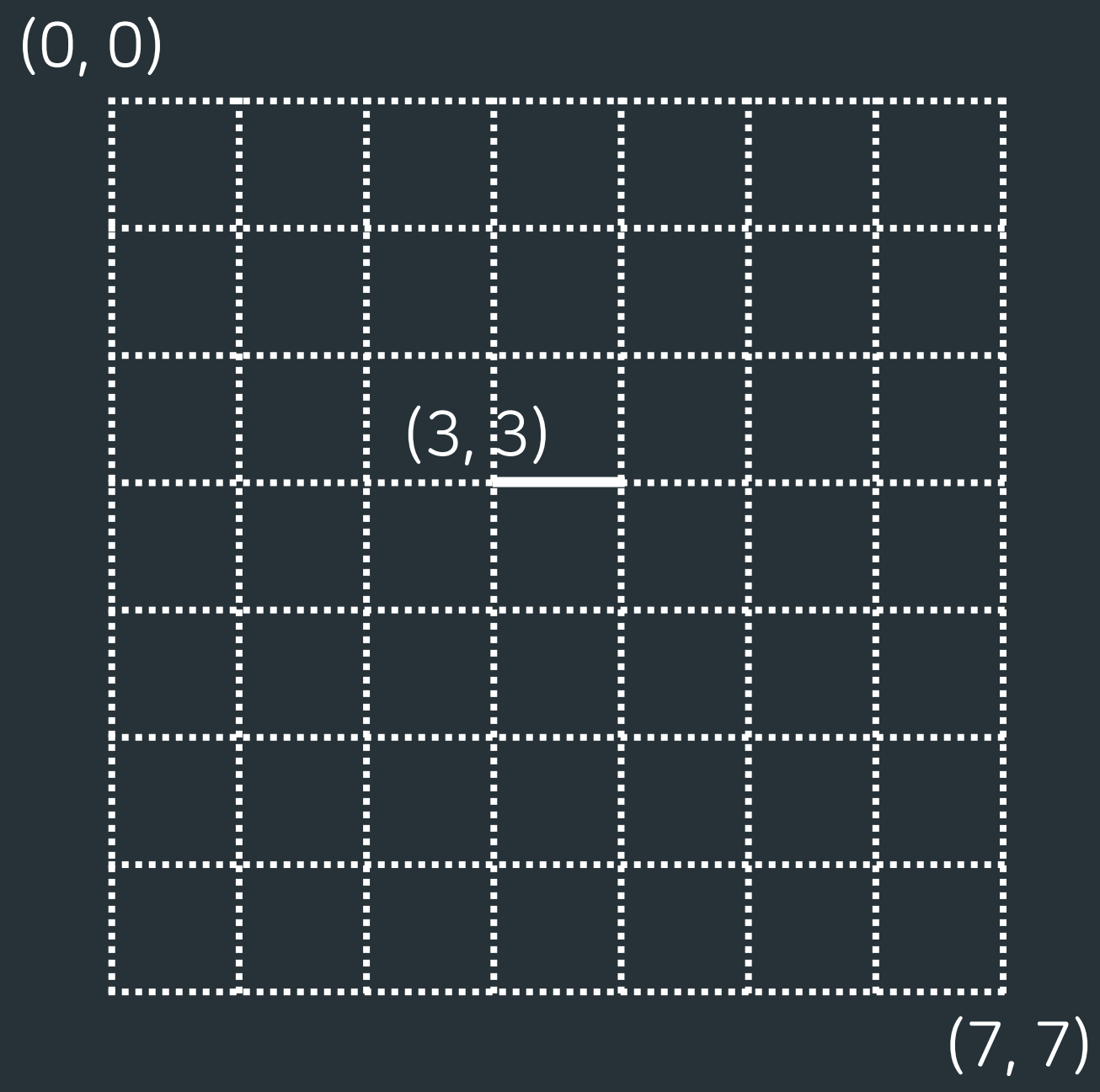


3세대

# 드래곤 커브를 그려보자



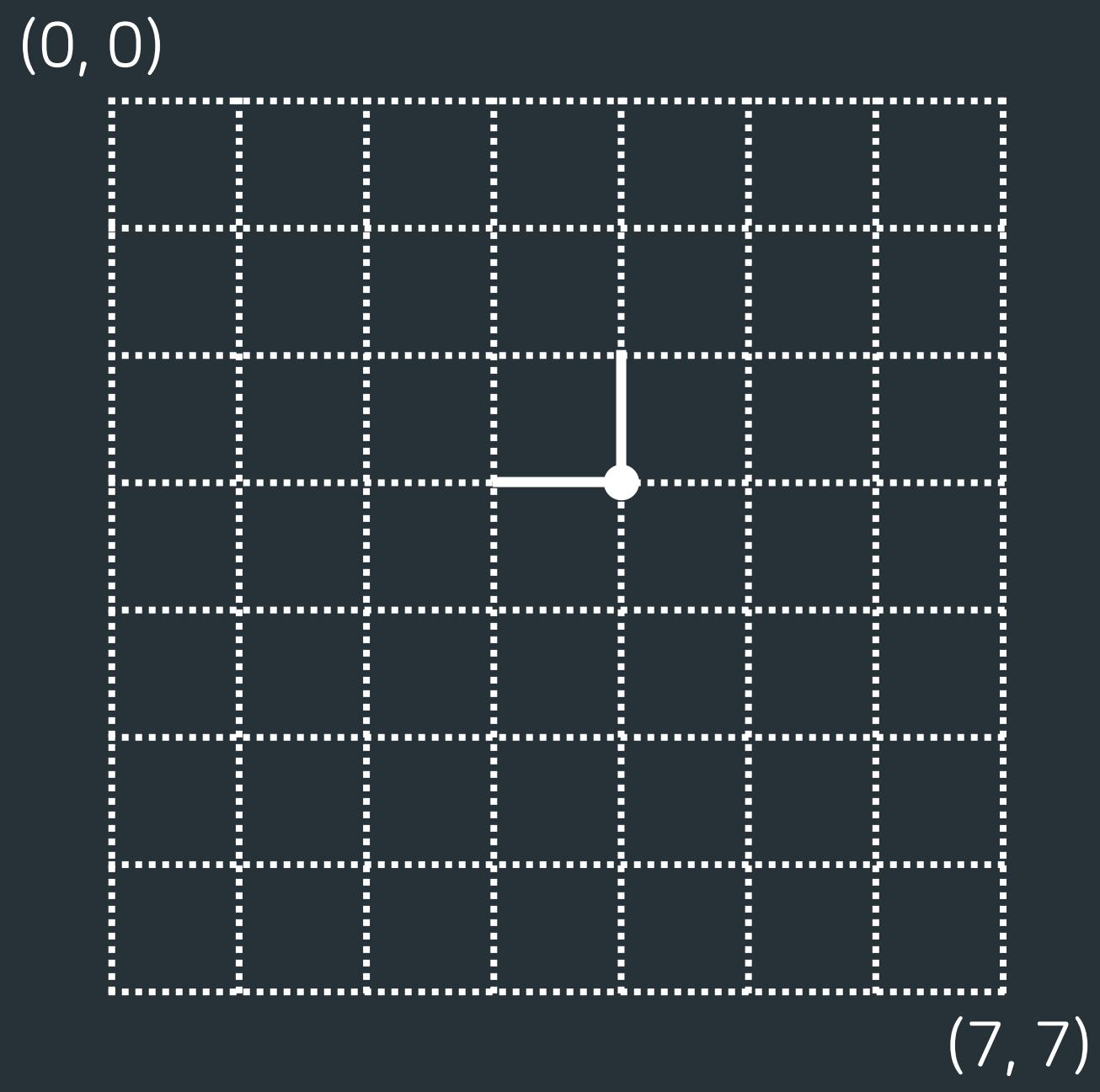
4세대



## 1번 커브 - 0세대

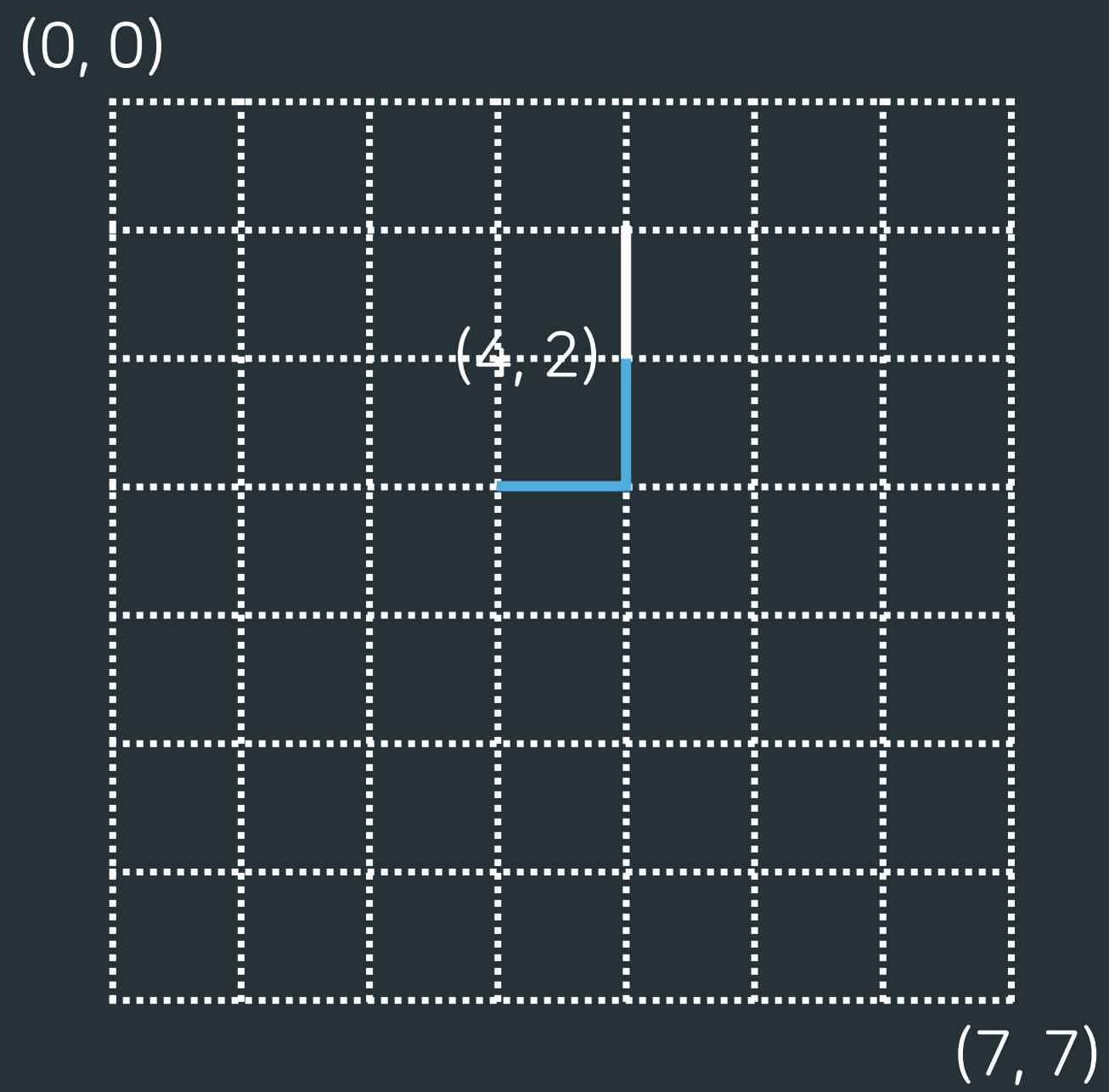
시작 점: (3, 3)  
시작 방향: 0  
세대: 1





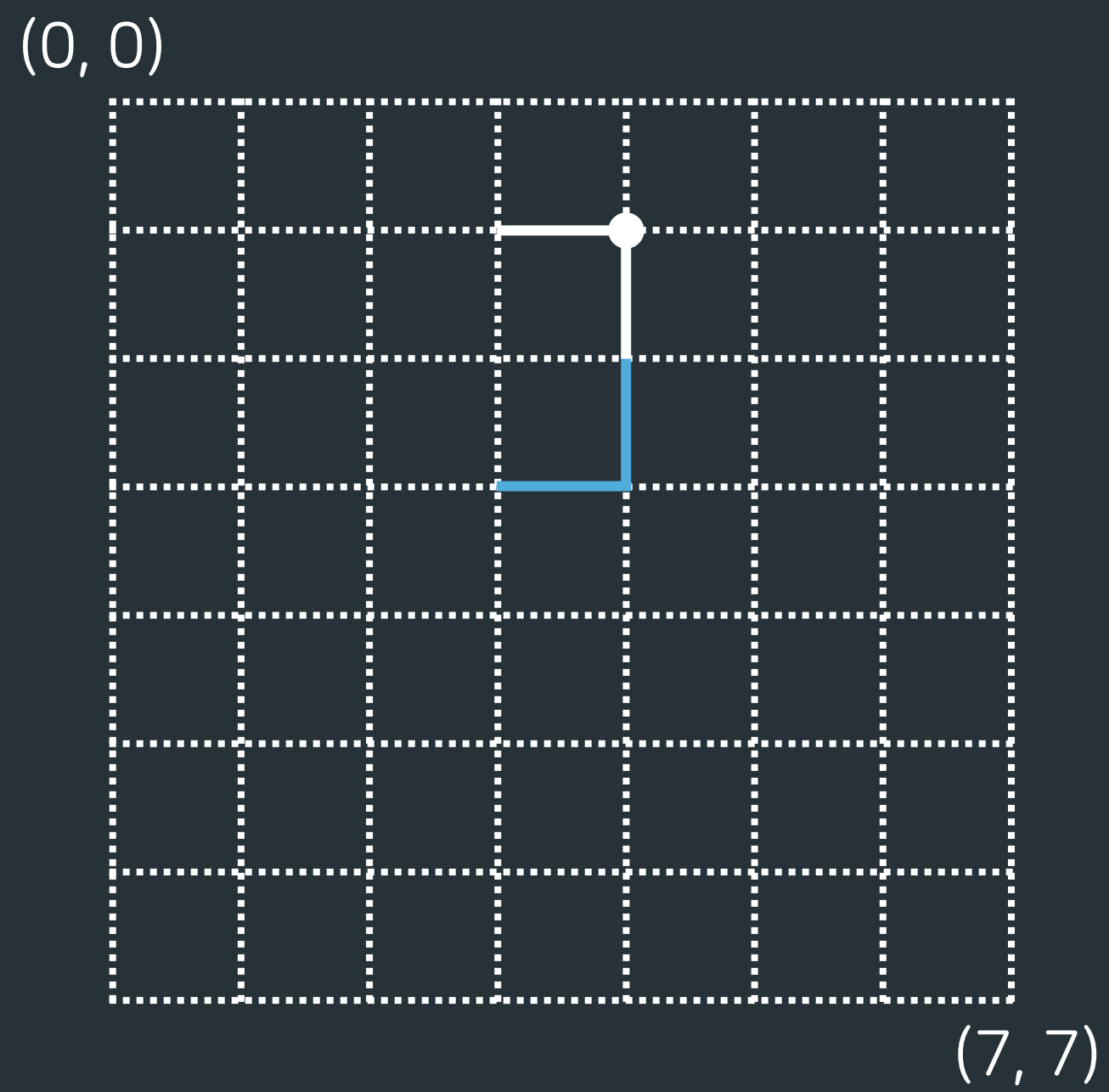
1번 커브 - 1세대

시작 점: (3, 3)  
시작 방향: 0  
세대: 1



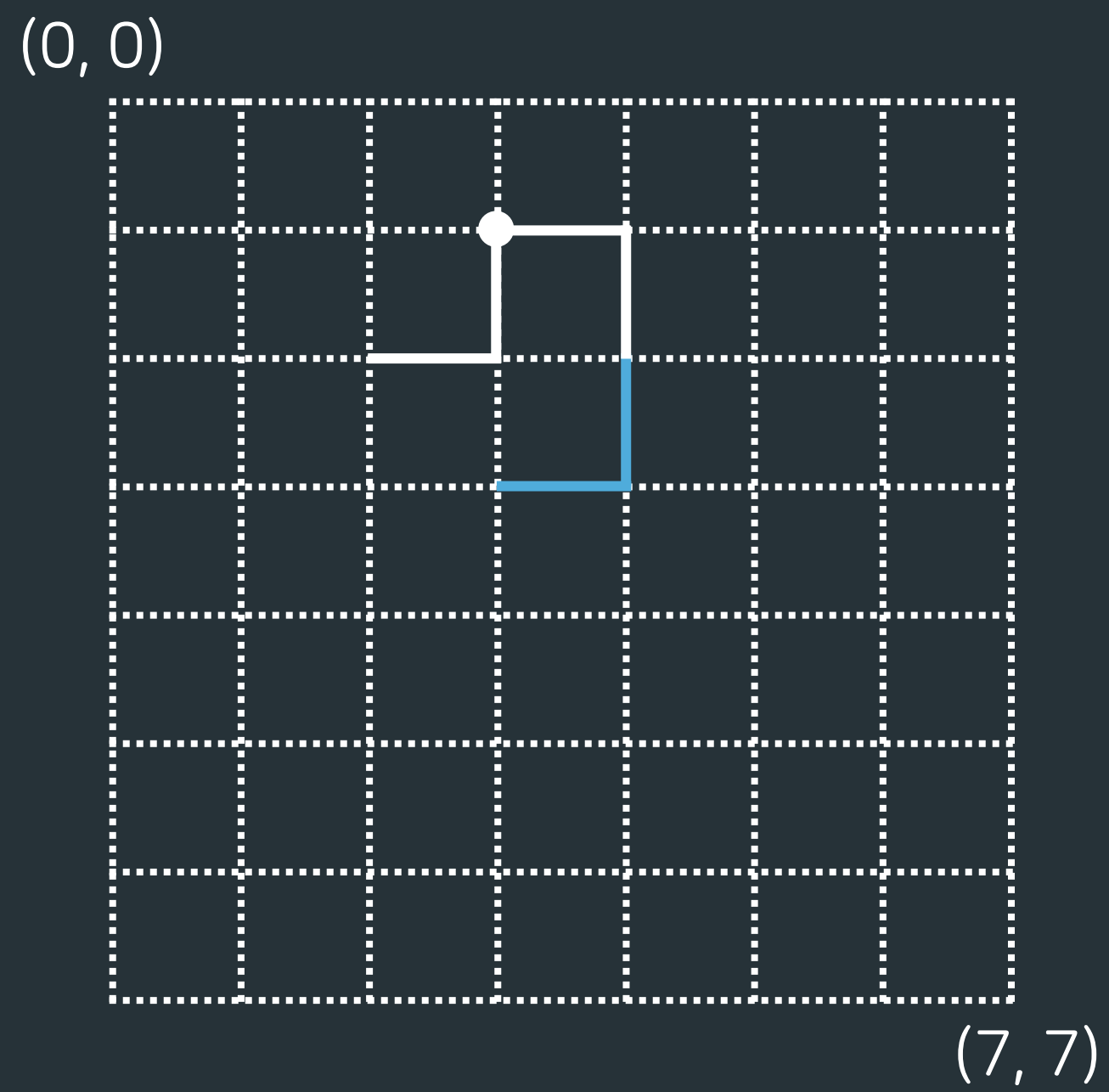
## 2번 커브 - 0세대

시작 점: (4, 2)  
시작 방향: 1  
세대: 3



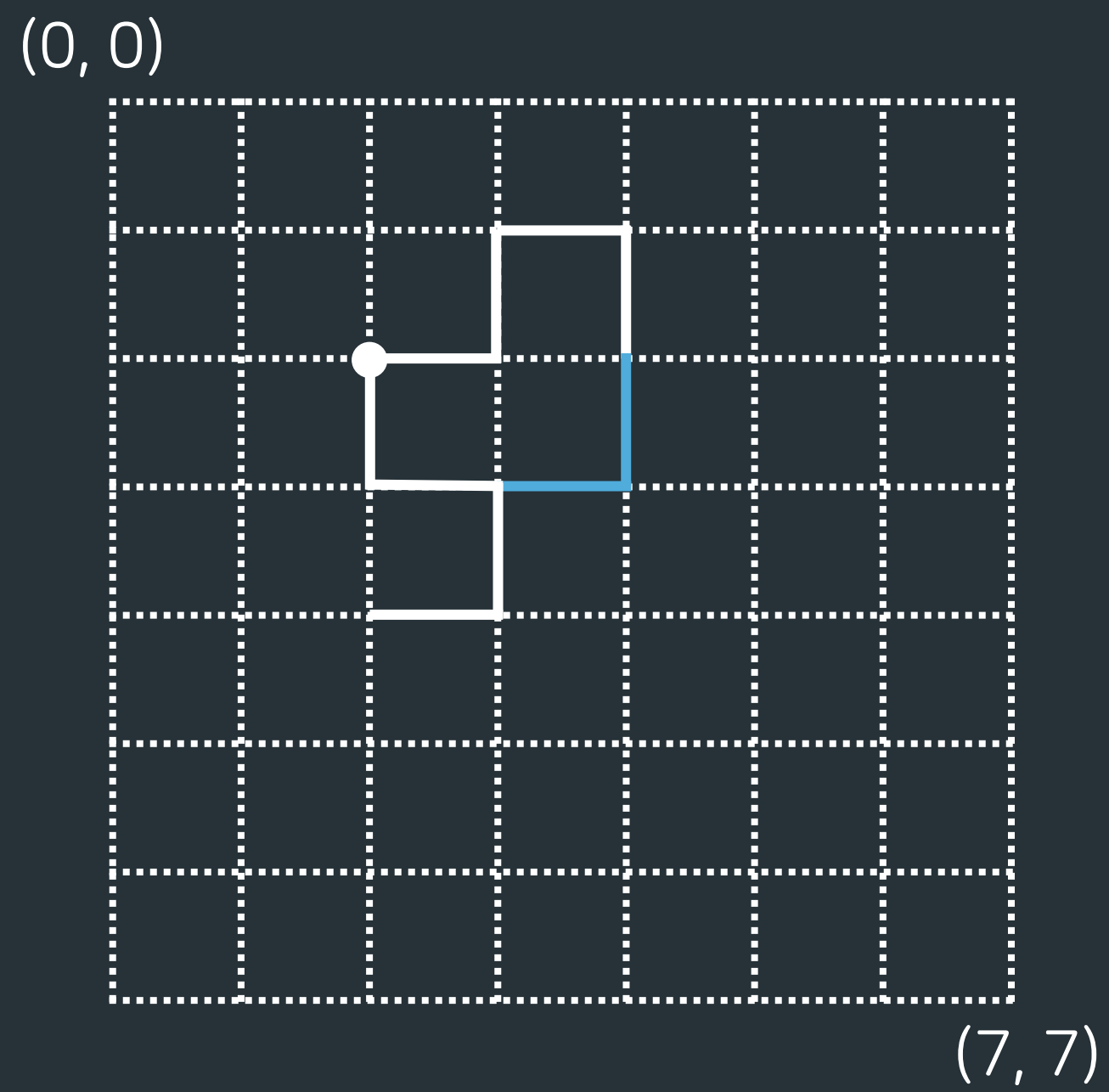
2번 커브 - 1세대

시작 점: (4, 2)  
시작 방향: 1  
세대: 3



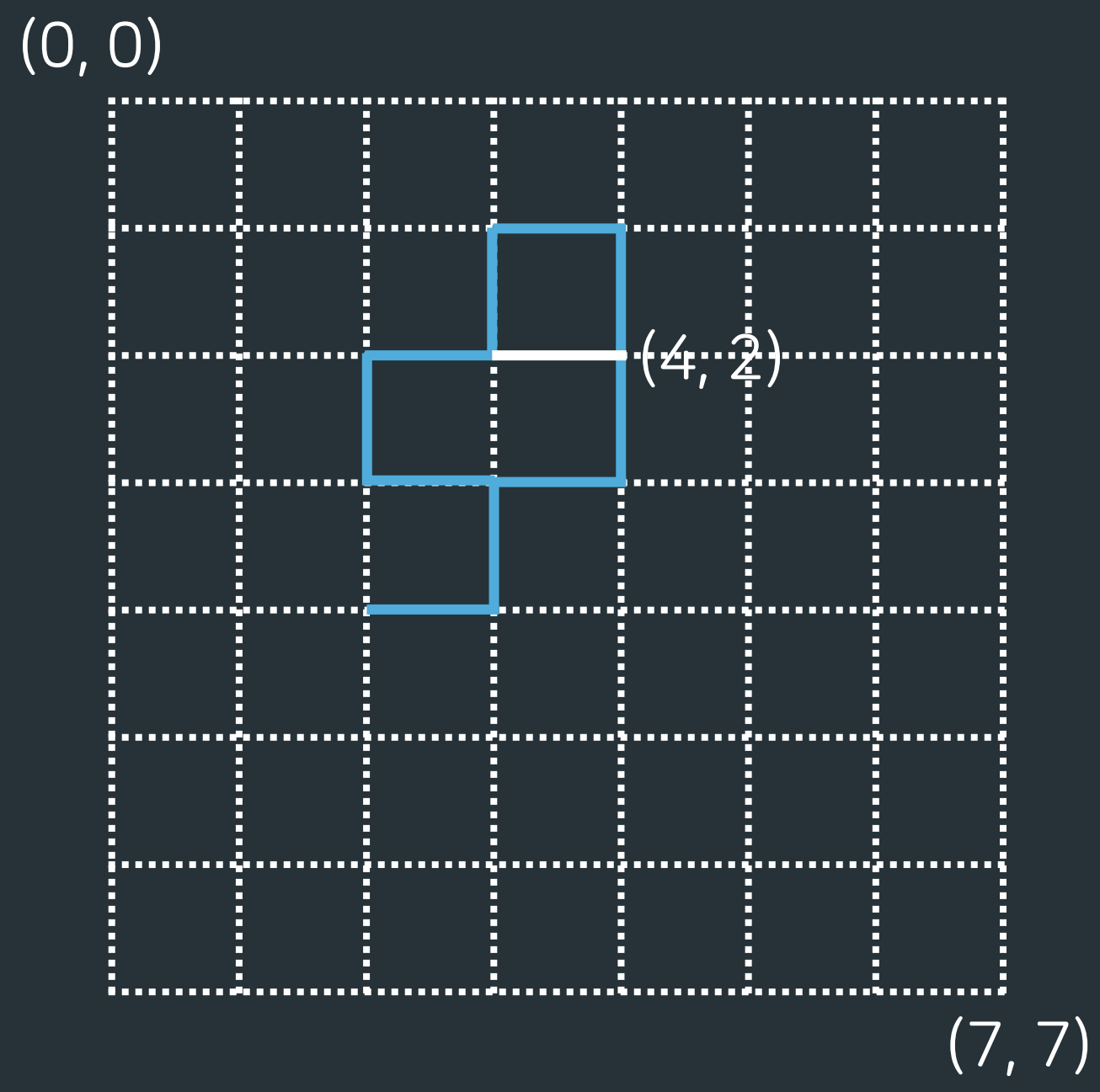
2번 커브 - 2세대

시작 점: (4, 2)  
시작 방향: 1  
세대: 3



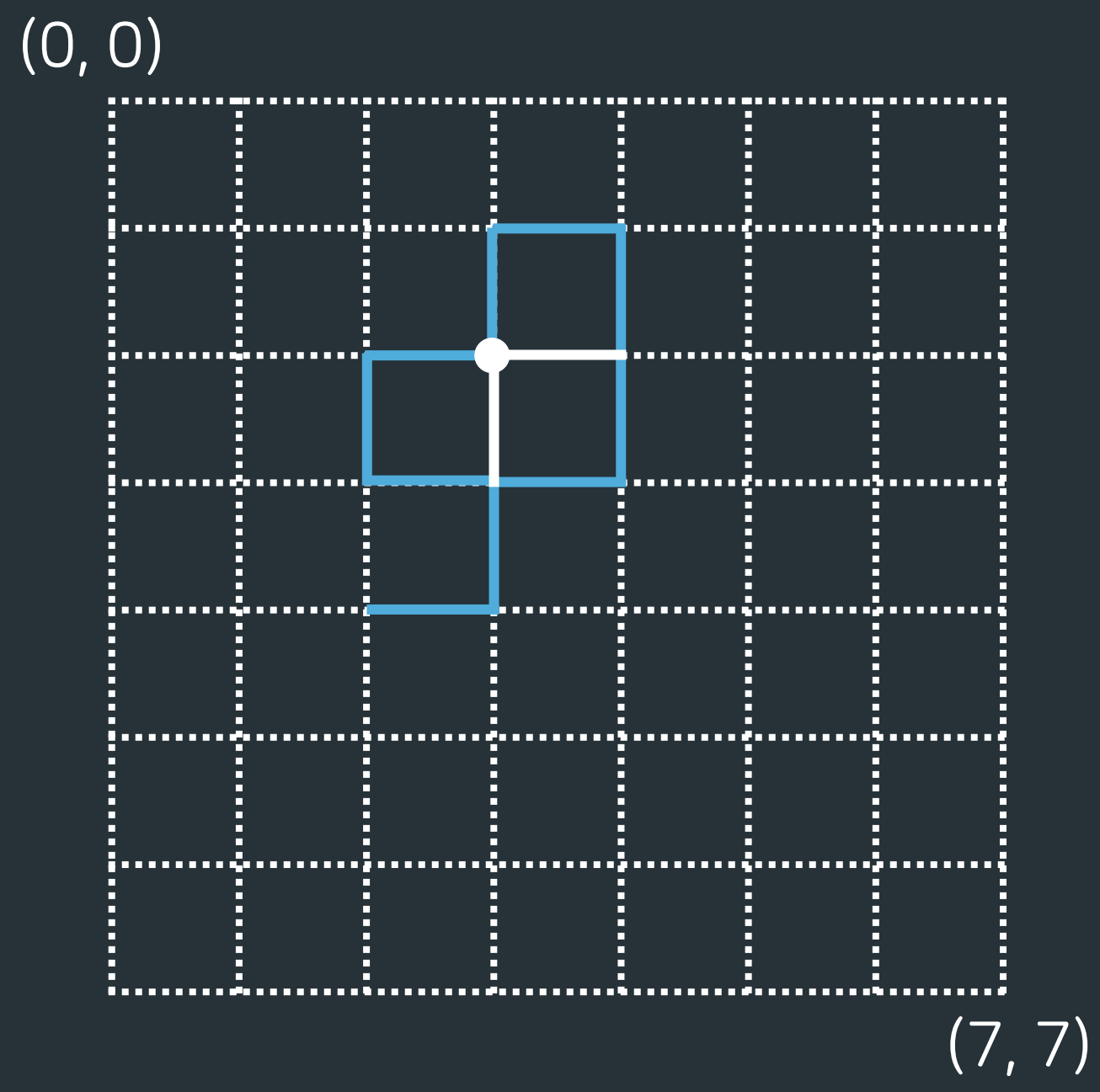
2번 커브 - 3세대

시작 점: (4, 2)  
시작 방향: 1  
세대: 3



## 3번 커브 - 0세대

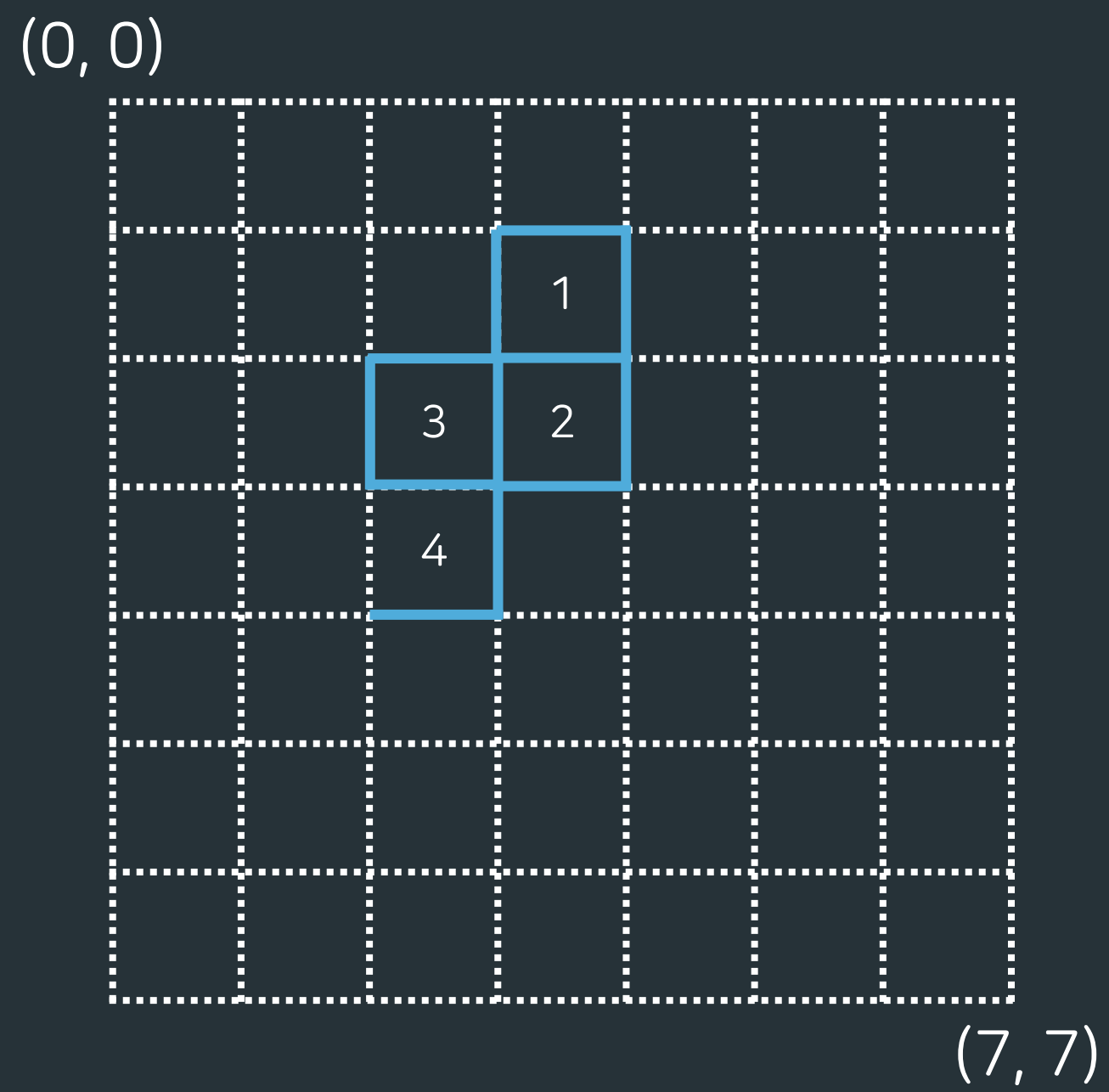
시작 점: (4, 2)  
시작 방향: 2  
세대: 1



3번 커브 - 1세대

시작 점: (4, 2)  
시작 방향: 2  
세대: 1

# 예제 1



정답: 4



## 좌표 VS 방향

## 좌표 VS 방향

다음 좌표 계산하기 VS 방향의 규칙성 찾기

# 규칙을 찾아보자



시계 방향 회전: 3 -> 2 -> 1 -> 0 -> 3 -> 2 -> 1 -> ...

0 세대: 0

1 세대: 0 1

2 세대: 0 1 2 1

3 세대: 0 1 2 1 2 3 2 1

...

N 세대: (N-1 세대) ((N-1 세대 거꾸로) + 1) -> 이대로 구현?

# 규칙을 찾아보자



시계 방향 회전: 3 -> 2 -> 1 -> 0 -> 3 -> 2 -> 1 -> ...

0 세대: 0

1 세대: 0 1

2 세대: 0 1 2 1

3 세대: 0 1 2 1 2 3 2 1

...

N 세대: (N-1 세대) (((N-1 세대 거꾸로) + 1) % 4)

평면(좌측 상단이  $(0, 0)$ )에 드래곤 커브를 그린 후 정사각형의 개수를 계산

1. 드래곤 커브는 평면 밖으로 나가지 않음으로 범위를 확인할 필요 없음
2. 0 세대의 드래곤 커브를 먼저 저장 (초기 조건)
3. 세대를 거듭하면서 드래곤 커브를 그림 (규칙을 파악하는 것이 중요)
4. 드래곤 커브가 그려진 평면 상의 정사각형의 개수 계산 (네 꼭짓점 확인)

추가로 풀어보면 좋은 문제!

/<> 1504번 : 특정한 최단 경로 - Gold 4

/<> 4485번 : 녹색 옷 입은 애가 젤다지? - Gold 4

/<> 10282번 : 해킹 - Gold 4

/<> 1613번 : 역사 - Gold 3